Cuadratic Calculation Algorithm with Optimized Stochasticity

C^2.A.O.S. Algorythm

Dhavit Prem (dhavitprem@gmail.com, yupanki@yupanainka.com)

Systems Engineer – Universidad de Lima

Philosophy of science and epistemology – Universidad Nacional Mayor de San Marcos

Abstract

This paper presents a new iterative method for calculating square roots that automatically converges to the exact value with adjustable precision. The algorithm, based on the iterative addition of proper fractions to the initial denominator, exhibits a self-correcting behavior that allows it to refine initial errors and achieve accurate results, regardless of the starting approximation used. This approach offers an innovative alternative to classical methods such as Newton-Raphson and Heron, standing out for its computational simplicity and numerical robustness. Furthermore, it is shown that the convergence is related to fixed-point theory, opening possibilities for its application in other contexts of numerical analysis. Theoretical and experimental tests are included to validate its effectiveness and efficiency.

Introduction

The calculation of square roots has been a fundamental need in mathematics since ancient times, with applications ranging from classical geometry to modern algorithms in computing and applied sciences. Throughout history, various methods have been developed to approximate square roots, from Babylonian approximations to the iterative algorithms of Heron and Newton-Raphson. While these methods are effective and widely used, they present certain limitations in terms of simplicity, robustness against initial errors, and ease of implementation in educational or low-computational-power environments.

In this context, the present work introduces an alternative approach for the iterative calculation of square roots based on the construction of proper fractions and a process of numerical self-correction. This method, distinguished by its self-regulated convergence and flexibility in the choice of initial conditions, offers a new perspective on iterative algorithms. Unlike other methods, it does not require an initial precise estimate nor the use of complex operations, making it an accessible tool both in educational and computational environments.

The significance of this method lies not only in its simplicity but also in its dynamic behavior, which converges to the exact value of the square root with a level of precision adjustable to the required number of decimal places. This process, intrinsically self-regulating, aligns with principles from fixed-point theory and opens new possibilities for understanding numerical algorithms with self-correction properties.

The aim of this paper is to present a detailed description of the method, demonstrate its theoretical and experimental convergence, and compare it with classical approaches to evaluate its advantages and limitations. Additionally, its potential applications in mathematical education, software design for calculations, and advanced numerical analysis will be explored.

Method Definition

The proposed method for calculating square roots is based on an iterative process that uses a sequence of proper fractions, where the denominator is self-regulated as the iterations progress.

Notation and Variable Description

This method proposes the calculation of square roots with the desired precision in decimal digits. It also introduces a user-friendly syntax (see Table 1) as follows:

Notation	**Interpretation (Formula)**
$\sqrt{N} = a \left\{ \dfrac{D}{2a*} \right\}$	$\sqrt{N} = a + \left\{ \dfrac{N - a^2}{2a + *} \right\}$

Where:

N is the number for which the square root is to be calculated.

a is the integer part of the square root of N, N, that is, the largest integer such that $a^2 < N$

D is the difference between N and a^2: $N - a^2$

$2a*$ is the denominator of the *generating fraction*, where:

> $2a$ (the double of a) is the integer part of the denominator of the *generating fraction*.
>
> ***** (asterisk) is the decimal part of the generating fraction, being a random number greater than zero and less than one (0 < ***** < 1) , with as many decimal places as the desired precision for the final square root calculation.

Important Clarification on Notation

- For an accurate interpretation of the notation, as indicated in the formula, it should be noted that:
- The result within the curly braces { } **is not multiplied**, but **added** to the value of a
- The value of *** is also not multiplied**, but **added** to the value of $2a$

Algoritmo de Cálculo Cuadrático con Aleatoriedad Optimizada y Sistemática (C^2.A.O.S.)

The proposed algorithm for calculating square roots is based on an iterative process that combines the introduction of a random value in the denominator with a successive refinement of the result. Despite the randomness introduced in the denominator, stable and precise convergence towards the exact value of the square root with the desired number of decimal places is observed, due to the following fundamental characteristics:

1. **Successive Refinement Iteration:** The algorithm employs an iterative approximation procedure, similar to the Newton-Raphson method for calculating square roots. In each iteration, the square root value is adjusted by correcting the denominator, which is influenced by a random decimal number. Despite the uncertainty introduced by the randomness, the structure of the algorithm ensures that the result progressively approaches the exact value of the square root.

2. **Convergence Towards the Exact Value:** As the iterations progress, the calculated value increasingly approximates the actual square root value. The nature of the iterative process allows the small variations introduced by the random value in the denominator not to prevent convergence. On the contrary, the algorithm is designed such that with each step, the difference between the calculated value and the actual value decreases in a controlled manner. The desired precision is achieved when the difference is smaller than a predefined threshold, ensuring that the calculated value will be sufficiently accurate for the specified number of digits.

3. **The Role of Random Chaos:** The introduction of a random number in the denominator, adjusted to as many digits as the desired precision, may seem like a source of chaos. However, the underlying mathematical structure of the algorithm ensures that this chaos self-regulates and ultimately leads to a precise result. This aligns with the principles of chaos theory, which suggests that nonlinear and chaotic systems can exhibit patterns and regularities over time. In this case, the "chaos" generated by the random number is not an obstacle, but rather a tool that introduces controlled variability into the process, allowing the algorithm to continuously adjust the approximation until the desired solution is reached.

4. **Precision Control:** The algorithm provides precise control over the number of decimal digits desired in the result. The random value in the denominator is adjusted to allow the necessary variability, but the number of iterations and the precision in the calculations ensure that the final square root value will be exact within the specified

precision limits. The flexibility of the algorithm allows square roots to be calculated with arbitrary precision while maintaining the stability and accuracy of the process.

Example of Application of Nomenclature and its Mathematical Interpretation

The notation equivalent to the square root of 13 is:

$$\sqrt{13} = 3\left\{\frac{4}{6*}\right\}$$

Its mathematical interpretation involves the application of the following formula:

$$\sqrt{N} = a + \left\{\frac{N - a^2}{2a + *}\right\} \rightarrow \sqrt{13} = 3 + \left\{\frac{13 - 3^2}{2 \cdot 3 + *}\right\} \rightarrow \sqrt{13} = 3 + \left\{\frac{4}{6 + *}\right\}$$

Algorithm Steps

Step 1: Initial approximation (Calculation of a)

The process begins with selecting an initial estimate for the square root of a number N. This estimate corresponds to the nearest integer square root to the actual value of $\sqrt{N}$, that is, the largest integer value a such that $a^2 \leq N$. This value, in addition to representing the integer part of the answer, will be used to calculate the first term of the fraction.

Example: $\sqrt{13} \rightarrow a = 3$

Step 2: Formation of the *Generating Fraction*

Once the initial estimate a has been obtained, we check if it is an exact square root ($a^2 = N$). In that case, the square root calculation will be complete, with a being the answer. Otherwise, to find the decimal part, the *generating fraction* must be formed.

Example: $\sqrt{13} \rightarrow$ N = 13; a = 3; * = 9778 (random number between 0 and 1)

- A random number is chosen that contains as many digits as the desired precision in the square root calculation.

Generating fraction formula		Replacing values		Obtained *generating fraction*
$\left\{\dfrac{\mathrm{N}-a^2}{2a+*}\right\}$	→	$\left\{\dfrac{13-3^2}{2\cdot 3+9778}\right\}$	→	$\left\{\dfrac{4}{6.9778}\right\}$

Step 3: Addition to the Denominator and Regeneration of the Generating Fraction

Once the value of the division of the generating fraction is obtained, it is rounded to the desired number of digits, and this result is added to the value of a to form the denominator of the next fraction to be calculated. The value of the numerator remains the same as at the beginning.

Example:

Generating fraction calculation		Denominator = a + value obtained		New generating fraction
$\left\{\dfrac{4}{6.9778}\right\} = 0.5732$	→	$\left\{\dfrac{4}{6+0.5732}\right\}$	→	$\left\{\dfrac{4}{6.5732}\right\}$

Step 4: Iteration of the Precision Adjustment Process

In each iteration, the procedure from Step 3 is repeated until one of the values obtained from dividing the fraction repeats. As the process progresses, the fraction converges or oscillates toward the exact value of the square root of N, with a precision determined by the number of digits initially assigned to the random variable $*$, which must match the digits used to round each iteration. The method guarantees an approximation of N with the desired precision.

Example:

1st iteration	2nd iteration	3th iteration
$\left\{\dfrac{4}{6.5732}\right\} = 0.6085$	$\left\{\dfrac{4}{6.6085}\right\} = 0.6053$	$\left\{\dfrac{4}{6.6053}\right\} = \mathbf{0.6056}$

4th iteration	5th iteration
$\left\{\dfrac{4}{6.6056}\right\} = 0.6055$	$\left\{\dfrac{4}{6.6055}\right\} = \mathbf{0.6056}$

Step 5: Addition of Integer and Decimal Parts

Once the repeating decimal is identified, it is added to the integer part, a.

Example: In this case, it corresponds: $3 + \left\{\frac{4}{6*}\right\} = 3 + 0.6056 = 3.6056$ (answer)

Proof of Algorithm Correctness

Step 1. Iterative formula

The iterative formula followed by the algorithm is:

$$\epsilon_{k+1} = \frac{N-a^2}{2a+\epsilon_k} = \left\{\frac{D}{2a*}\right\}$$

Where:

ϵ_{k+1} is the decimal fraction of the denominator of the generating fraction in iteration *k+1*

N is the number for which the square root is to be calculated.

a is the integer part of the square root of N, the largest integer such that $a^2 < N$

D is the difference between N and the closest perfect square below it, $N - a^2$

ϵ_k is the decimal part of the generating fraction denominator in iteration k / $(\epsilon_k \neq 0 \wedge \epsilon_k \neq 1)$

Step 2. Establishment of Convergence

Analyzing the convergence of the sequence of $\left\{\frac{D}{2a*}\right\}$ towards $\sqrt{N} - a$

2.1 Error condition in the approximation: we define the error in the *k-th* iteration as:

$$\epsilon_{k+1} = \sqrt{N} - a_{k+1} \qquad \text{... } \textbf{\textit{Eq. 1}}$$

Applying the iterative formula, the error in the next iteration ϵ_{k+1} is:

$$\epsilon_{k+1} = \sqrt{N} - a_{k+1} = \sqrt{N} - \left(a_k + \frac{N-a_k^2}{2a_k+\epsilon_k}\right) \qquad \text{... Eq. 2}$$

We replace a_k with $\sqrt{N} - \epsilon_k$:

$$\epsilon_{k+1} = \sqrt{N} - \left(\sqrt{N} - \epsilon_k + \frac{N-(\sqrt{N}-\epsilon_k)^2}{2(\sqrt{N}-\epsilon_k) + \epsilon_k} \right) \qquad \text{... Eq. 3}$$

Simplifying we obtain:

$$\epsilon_{k+1} = \epsilon_k - \frac{N-(\sqrt{N}-\epsilon_k)^2}{2(\sqrt{N}-\epsilon_k) + \epsilon_k} \qquad \text{... Eq.4}$$

2.2 Expansion for small errors:

Assuming that ϵ_k is small, which occurs when *k* is large and a_k approaches $\sqrt{N}$, we can expand the quadratic term as follows:

$$(\sqrt{N} - \epsilon_k)^2 = N - 2\sqrt{N}\epsilon_k + \epsilon_k^2 \qquad \text{... Eq. 5}$$

Thus, the expression for the error becomes:

$$\epsilon_{k+1} = \epsilon_k - \frac{N-(N - 2\sqrt{N}\epsilon_k + \epsilon_k^2)}{2(\sqrt{N}-\epsilon_k) + \epsilon_k} \qquad \text{... Eq. 6}$$

$$\epsilon_{k+1} = \epsilon_k - \frac{(2\sqrt{N}\epsilon_k - \epsilon_k^2)}{2(\sqrt{N}-\epsilon_k) + \epsilon_k} \qquad \text{... Eq. 7}$$

2.3 Aproximation for small errors:

If ϵ_k is sufficiently small, we can approximate the denominator sa $2\sqrt{N}$, yielding:

$$\epsilon_{k+1} \approx \epsilon_k - \frac{2\sqrt{N}\ \epsilon_k}{2\sqrt{N}} \approx \epsilon_k - \epsilon_k \approx 0 \qquad \text{... Eq. 8}$$

Thus, the error decreases with each iteration. This implies that, as the iterations progress, the error is reduced.

Step 3. Establishment of convergence

The quadratic convergence of an iterative method means that the error in the *(k+1)th* iteration decreases proportionally to the square of the error in the *kth* iteration. In this case, the form of the iteration suggests that the convergence rate will be quadratic, ensuring that the error reduces rapidly with each iteration.

That is, we can expect the error to decrease by a factor of approximately e_k^2 in each iteration, which implies quite fast convergence

Step 4. Influence of ϵ_k

The initial random value ϵ_k has minimal influence on the convergence of the algorithm. Since it is in the denominator, and as the value of a_k approaches $\sqrt{N}$ more and more, the term ϵ_k becomes increasingly insignificant as the iterations progress. In other words, the contribution of ϵ_k does not affect long-term convergence because it is controlled by the structure of the iteration, similar to the Newton-Raphson formula, which is known for its quadratic convergence

Conclusions

The square root calculation algorithm described in this study demonstrates that, even from a source of randomness (the random number in the denominator), it is possible to achieve exact precision through an iterative process that continuously refines the approximate value. The combination of controlled chaos and successive convergence allows this method to be robust and efficient in calculating square roots with the desired precision, regardless of the initial random fluctuations.

The algebraic demonstration we propose shows that the algorithm will always converge quickly and precisely towards $\sqrt{N}$. The randomness introduced by ϵ_k does not hinder the convergence but only introduces a temporary fluctuation that does not affect the overall behavior of the algorithm, which follows a scheme similar to the Newton-Raphson formula.

Therefore, the algorithm is robust and will always ensure convergence to the exact square root with the desired precision.

The proposed nomenclature is recommended, as the format facilitates the representation and understanding of this iterative square root calculation method. This notation not only optimizes the presentation of calculations but also adds remarkable mathematical elegance. As observed in the table of square roots (see table 1), the proposed notation enables the appreciation of a harmonic sequence in the various expressions. This stylized approach enhances both the presentation and clarity of the underlying algorithm, offering a more accessible and comprehensible way to visualize the convergence of results with the desired precision. Ultimately, adopting this nomenclature provides both clarity and aesthetic value to the mathematics involved in calculating square roots.

Table 1. Table of square roots – Suggested nomenclature

$\sqrt{1} = 1$	$\sqrt{26} = 5\left\{\frac{1}{10*}\right\}$	$\sqrt{51} = 7\left\{\frac{2}{14*}\right\}$	$\sqrt{76} = 8\left\{\frac{12}{16*}\right\}$
$\sqrt{2} = 1\left\{\frac{1}{2*}\right\}$	$\sqrt{27} = 5\left\{\frac{2}{10*}\right\}$	$\sqrt{52} = 7\left\{\frac{3}{14*}\right\}$	$\sqrt{77} = 8\left\{\frac{13}{16*}\right\}$
$\sqrt{3} = 1\left\{\frac{2}{2*}\right\}$	$\sqrt{28} = 5\left\{\frac{3}{10*}\right\}$	$\sqrt{53} = 7\left\{\frac{4}{14*}\right\}$	$\sqrt{78} = 8\left\{\frac{14}{16*}\right\}$
$\sqrt{4} = 2$	$\sqrt{29} = 5\left\{\frac{4}{10*}\right\}$	$\sqrt{54} = 7\left\{\frac{5}{14*}\right\}$	$\sqrt{79} = 8\left\{\frac{15}{16*}\right\}$
$\sqrt{5} = 2\left\{\frac{1}{4*}\right\}$	$\sqrt{30} = 5\left\{\frac{5}{10*}\right\}$	$\sqrt{55} = 7\left\{\frac{6}{14*}\right\}$	$\sqrt{80} = 8\left\{\frac{16}{16*}\right\}$
$\sqrt{6} = 2\left\{\frac{2}{4*}\right\}$	$\sqrt{31} = 5\left\{\frac{6}{10*}\right\}$	$\sqrt{56} = 7\left\{\frac{7}{14*}\right\}$	$\sqrt{81} = 9$
$\sqrt{7} = 2\left\{\frac{3}{4*}\right\}$	$\sqrt{32} = 5\left\{\frac{7}{10*}\right\}$	$\sqrt{57} = 7\left\{\frac{8}{14*}\right\}$	$\sqrt{82} = 9\left\{\frac{1}{18*}\right\}$
$\sqrt{8} = 2\left\{\frac{4}{4*}\right\}$	$\sqrt{33} = 5\left\{\frac{8}{10*}\right\}$	$\sqrt{58} = 7\left\{\frac{9}{14*}\right\}$	$\sqrt{83} = 9\left\{\frac{2}{18*}\right\}$
$\sqrt{9} = 3$	$\sqrt{34} = 5\left\{\frac{9}{10*}\right\}$	$\sqrt{59} = 7\left\{\frac{10}{14*}\right\}$	$\sqrt{84} = 9\left\{\frac{3}{18*}\right\}$
$\sqrt{10} = 3\left\{\frac{1}{6*}\right\}$	$\sqrt{35} = 5\left\{\frac{10}{10*}\right\}$	$\sqrt{60} = 7\left\{\frac{11}{14*}\right\}$	$\sqrt{85} = 9\left\{\frac{4}{18*}\right\}$
$\sqrt{11} = 3\left\{\frac{2}{6*}\right\}$	$\sqrt{36} = 6$	$\sqrt{61} = 7\left\{\frac{12}{14*}\right\}$	$\sqrt{86} = 9\left\{\frac{5}{18*}\right\}$
$\sqrt{12} = 3\left\{\frac{3}{6*}\right\}$	$\sqrt{37} = 6\left\{\frac{1}{12*}\right\}$	$\sqrt{62} = 7\left\{\frac{13}{14*}\right\}$	$\sqrt{87} = 9\left\{\frac{6}{18*}\right\}$
$\sqrt{13} = 3\left\{\frac{4}{6*}\right\}$	$\sqrt{38} = 6\left\{\frac{2}{12*}\right\}$	$\sqrt{63} = 7\left\{\frac{14}{14*}\right\}$	$\sqrt{88} = 9\left\{\frac{7}{18*}\right\}$
$\sqrt{14} = 3\left\{\frac{5}{6*}\right\}$	$\sqrt{39} = 6\left\{\frac{3}{12*}\right\}$	$\sqrt{64} = 8$	$\sqrt{89} = 9\left\{\frac{8}{18*}\right\}$
$\sqrt{15} = 3\left\{\frac{6}{6*}\right\}$	$\sqrt{40} = 6\left\{\frac{4}{12*}\right\}$	$\sqrt{65} = 8\left\{\frac{1}{16*}\right\}$	$\sqrt{90} = 9\left\{\frac{9}{18*}\right\}$
$\sqrt{16} = 4$	$\sqrt{41} = 6\left\{\frac{5}{12*}\right\}$	$\sqrt{66} = 8\left\{\frac{2}{16*}\right\}$	$\sqrt{91} = 9\left\{\frac{10}{18*}\right\}$
$\sqrt{17} = 4\left\{\frac{1}{8*}\right\}$	$\sqrt{42} = 6\left\{\frac{6}{12*}\right\}$	$\sqrt{67} = 8\left\{\frac{3}{16*}\right\}$	$\sqrt{92} = 9\left\{\frac{11}{18*}\right\}$
$\sqrt{18} = 4\left\{\frac{2}{8*}\right\}$	$\sqrt{43} = 6\left\{\frac{7}{12*}\right\}$	$\sqrt{68} = 8\left\{\frac{4}{16*}\right\}$	$\sqrt{93} = 9\left\{\frac{12}{18*}\right\}$
$\sqrt{19} = 4\left\{\frac{3}{8*}\right\}$	$\sqrt{44} = 6\left\{\frac{8}{12*}\right\}$	$\sqrt{69} = 8\left\{\frac{5}{16*}\right\}$	$\sqrt{94} = 9\left\{\frac{13}{18*}\right\}$
$\sqrt{20} = 4\left\{\frac{4}{8*}\right\}$	$\sqrt{45} = 6\left\{\frac{9}{12*}\right\}$	$\sqrt{70} = 8\left\{\frac{6}{16*}\right\}$	$\sqrt{95} = 9\left\{\frac{14}{18*}\right\}$
$\sqrt{21} = 4\left\{\frac{5}{8*}\right\}$	$\sqrt{46} = 6\left\{\frac{10}{12*}\right\}$	$\sqrt{71} = 8\left\{\frac{7}{16*}\right\}$	$\sqrt{96} = 9\left\{\frac{15}{18*}\right\}$
$\sqrt{22} = 4\left\{\frac{6}{8*}\right\}$	$\sqrt{47} = 6\left\{\frac{11}{12*}\right\}$	$\sqrt{72} = 8\left\{\frac{8}{16*}\right\}$	$\sqrt{97} = 9\left\{\frac{16}{18*}\right\}$
$\sqrt{23} = 4\left\{\frac{7}{8*}\right\}$	$\sqrt{48} = 6\left\{\frac{12}{12*}\right\}$	$\sqrt{73} = 8\left\{\frac{9}{16*}\right\}$	$\sqrt{98} = 9\left\{\frac{17}{18*}\right\}$
$\sqrt{24} = 4\left\{\frac{8}{8*}\right\}$	$\sqrt{49} = 7$	$\sqrt{74} = 8\left\{\frac{10}{16*}\right\}$	$\sqrt{99} = 9\left\{\frac{18}{18*}\right\}$
$\sqrt{25} = 5$	$\sqrt{50} = 7\left\{\frac{1}{14*}\right\}$	$\sqrt{75} = 8\left\{\frac{11}{16*}\right\}$	$\sqrt{100} = 10$

Bibliography

Atkinson, K. E. (2008). An introduction to numerical analysis (2nd ed.). Wiley.

Burden, R. L., & Faires, J. D. (2010). *Numerical analysis* (9ª ed.). Cengage Learning.

Kincaid, D., & Cheney, W. (2002). Numerical analysis: Mathematics of scientific computing (3rd ed.). Brooks/Cole.

Dahlquist, G., & Bjorck, Å. (2003). Numerical methods for scientists and engineers (2nd ed.). Dover Publications.